AMENDMENTS TO THE CLAIMS

Please amend the claims as follows:

1. (Currently amended) A method comprising:

determining a new slack value based on current resource constraints, for each of one or more ready instructions in a scheduling region;

selecting one of the ready instructions, based on the new slack value of the one ready instruction;

scheduling the selected ready instruction; and

repeating the method for determining, selecting and scheduling for each of the one or more ready instructions remaining to be selected and scheduled until all ready instructions have been scheduled.

2. (Canceled)

3. (Currently amended) The method of claim 1, wherein determining the new slack value comprises:

determining the new slack value for each of the one or more ready instructions based on resource constraints and dependence height.

4. (Currently amended) The method of claim 1, wherein determining the new slack value comprises:

determining a dependence deadline based on a dependence height for each of the one or more ready instructions;

determining a resource deadline based on resource constraints for each of the one or more ready instructions;

selecting as a deadline value that indicates a least number of cycles, between the resource deadline and the dependence deadline; and

determining the new slack value based on the selected deadline value.

5. (Currently amended) The method of claim 1, wherein selecting one of the ready instructions comprises selecting the ready instruction having a lowest new slack value.

6. (Original) The method of claim 1, further comprising:

generating an entry in a ready list for each of the one or more ready instructions; and

removing the entry for the selected ready instruction from the ready list.

7. (Currently amended) The method of claim 6, further comprising:

adding to an uncover list any non-ready instructions uncovered by the scheduling of the selected ready instruction, wherein the non-ready instructions are dependent on the selected ready instruction.

8. (Previously presented) The method of claim 6, further comprising:

advancing a virtual clock to a subsequent clock cycle when there are no ready instructions in the ready list that can be scheduled in a clock cycle; and

adding an entry to the ready list for any non-ready instruction that becomes ready

in the subsequent clock cycle.

9. (Currently amended) The method of claim 4, wherein determining the new slack value comprises:

determining a minimum number of cycles needed to schedule each of the one or more ready instructions in the scheduling region, taking resource constraints into account;

determining the dependence deadline based on the dependence height and the minimum number of cycles; and

determining the resource deadline based on resource constraints and the minimum number of cycles.

10. (Previously presented) The method of claim 9, wherein determining the minimum number of cycles comprises:

determining a dependence length of the scheduling region;

determining a resource length of the scheduling region;

assigning the dependence length as the minimum number of cycles when the dependence length is greater than the resource length; and

assigning the resource length as the minimum number of cycles when the resource length is greater than the dependence length.

11. (Original) The method of claim 10, further comprising:

calculating the dependence length of the scheduling region based on the total height of a dependence graph of the scheduling region; and

calculating the resource length of the scheduling region based on the maximum number of cycles needed to schedule the instructions of the scheduling region for a machine resource.

12. (Previously presented) The method of claim 1, wherein the resource constraints comprise the maximum number of instructions of a particular instruction type that can be scheduled during a given cycle for a target processor.

13. (Currently amended) An article comprising:

a computer readable medium having a plurality of machine accessible instructions stored thereon, which when executed by a computer, cause the computer to perform the following method:

determining a new slack value based on resource constraints, for each of one or more ready instructions in a scheduling region;

selecting one of the ready instructions, based on the new slack value of the one ready instruction;

scheduling the selected ready instruction; and

repeating the method for determining, selecting and scheduling for each of the one or more ready instructions remaining to be selected and scheduled until all ready instructions have been scheduled.

14. (Canceled)

15. (Currently amended) The medium of claim 13, wherein determining the new slack value comprises:

determining the new slack value for each of the one or more ready instructions based on resource constraints and dependence height.

16. (Currently amended) The medium of claim 13, wherein determining the new slack value comprises:

determining a dependence deadline based on a dependence height for each of the one or more ready instructions;

determining a resource deadline based on resource constraints for each of the one or more ready instructions;

selecting as a deadline value that indicates a least number of cycles, between the resource deadline and the dependence deadline; and

determining the new slack value based on the selected deadline value.

17. (Previously presented) The medium of claim 13, wherein selecting one of the ready instructions comprises selecting a ready instruction having a highest scheduling priority.

18. (Previously presented) The medium of claim 13, further comprising:

generating an entry in a ready list for each of the one or more ready instructions; and

removing the entry for the selected ready instruction from the ready list.

19. (Currently amended) The medium of claim 18, further comprising:

adding to an uncover list any non-ready instructions uncovered by the scheduling of the selected ready instruction, wherein the non-ready instructions are dependent on the selected ready instruction.

20. (Previously presented) The medium of claim 18, further comprising:

advancing a virtual clock to a subsequent clock cycle when there are no ready instructions in the ready list that can be scheduled in a clock cycle; and

adding an entry to the ready list for any non-ready instruction that becomes ready in the subsequent clock cycle.

21. (Currently amended) The medium of claim 16, wherein determining the new slack value comprises:

determining a minimum number of cycles needed to schedule each of the one or more ready instructions in the scheduling region, taking resource constraints into account;

determining the dependence deadline based on the dependence height and the minimum number of cycles; and

determining the resource deadline based on resource constraints and the minimum number of cycles.

22. (Currently amended) The medium of claim 21, further wherein determining the minimum number of cycles comprises:

determining a dependence length of the scheduling region;

determining a resource length of the scheduling region;

assigning the dependence length as the minimum number of cycles when the dependence length is greater than the resource length; and

assigning the resource length as the minimum number of cycles when the resource length is greater than the dependence ~~length;~~ length.

23. (Previously presented) The medium of claim 22, further comprising:

calculating the dependence length of the scheduling region based on the total height of a dependence graph of the scheduling region; and

calculating the resource length of the scheduling region based on the maximum number of cycles needed to schedule the instructions of the scheduling region for a machine resource.

24. (Previously presented) The medium of claim 13, wherein the resource constraints comprise the maximum number of instructions of a particular instruction type that can be scheduled during a given cycle for a target processor.

25. (Currently amended) ~~A compiler~~ An apparatus for compiling a high-level programming language into an object code comprising:

a front end to receive a source code; and

a code generator, coupled to the front end, to:

receive the source code from the front end; and

compile the received source code into ~~an~~ the object code,

wherein the code generator includes one or more resource-aware schedulers to:

determine a new slack value based on current resource constraints, for each of one or more ready instructions in a scheduling region;

select one of the ready instructions, based on the new slack value of the one ready instruction;

schedule the selected ready instruction; and

repeat the method for determining, selecting and scheduling for each of the one or more ready instructions remaining to be selected and scheduled until all ready instructions have been scheduled.

26. (Currently amended) The ~~compiler~~ apparatus of claim 25, wherein the one or more resource-aware schedulers are to:

determine a first scheduling deadline for ~~for~~ each of the one or more ready instructions in the scheduling region, taking dependence considerations into account;

determine a second scheduling deadline for ~~for~~ each of the one or more ready instructions, taking resource constraints into account; and

select as a scheduling priority for each of the one or more ready instructions, between the first and second scheduling deadlines.

27. (Canceled)

28. (Currently amended) The ~~compiler~~ apparatus of claim 26, wherein the one or more resource-aware schedulers are to select the instruction for scheduling based on its scheduling priority.

29. (Currently amended) The ~~compiler~~ apparatus of claim 25, wherein the resource constraints comprise a maximum number of instructions that can be scheduled per cycle.

30. (Currently amended) The ~~compiler~~ apparatus of claim 25, wherein the resource constraints include the maximum number of instructions of a particular instruction type that can be scheduled per cycle.

31. (Currently amended) The ~~compiler~~ apparatus of claim 25, wherein the one or more resource-aware schedulers are to schedule the instructions such that instructions of a particular instruction type are distributed evenly among two or more resources.

32. (Currently amended) A system comprising:

a processor to execute each of one or more ready instructions; and

a memory system, coupled to the processor, to store each of the one or more ready instructions;

wherein the instructions include a resource-aware scheduler to:

determine a new slack value based on current resource constraints, for each of the one or more ready instructions in a scheduling region;

select one of the ready instructions, based on the new slack value of the

one ready instruction;

schedule the selected ready instruction; and

repeat the method for determining, selecting and scheduling for each of the one or more ready instructions remaining to be selected and scheduled until all ready instructions have been scheduled.

33. (Currently amended) The system of claim 32, wherein[[:]] the memory system ~~includes~~ comprises a Dynamic Random Access Memory (DRAM).

34. (Previously presented) The system of claim 32, wherein the resource-aware scheduler is to:

determine a first scheduling deadline for each of the one or more ready instructions in the scheduling region, taking dependence considerations into account;

determine a second scheduling deadline for each of the one or more ready instructions, taking resource constraints into account; and

select a scheduling priority for the instruction, between the first and second scheduling deadlines.

35. (Canceled)

36. (Previously presented) The system of claim 34, wherein the resource-aware scheduler is to select the instruction for scheduling based on its scheduling priority.

37. (Currently amended) The system of claim 32, wherein the resource constraints ~~include~~ comprise a maximum number of instructions that can be scheduled per cycle.

38. (Currently amended) The system of claim 32, wherein the resource constraints ~~include~~ comprise the maximum number of instructions of a particular instruction type that can be scheduled per cycle.